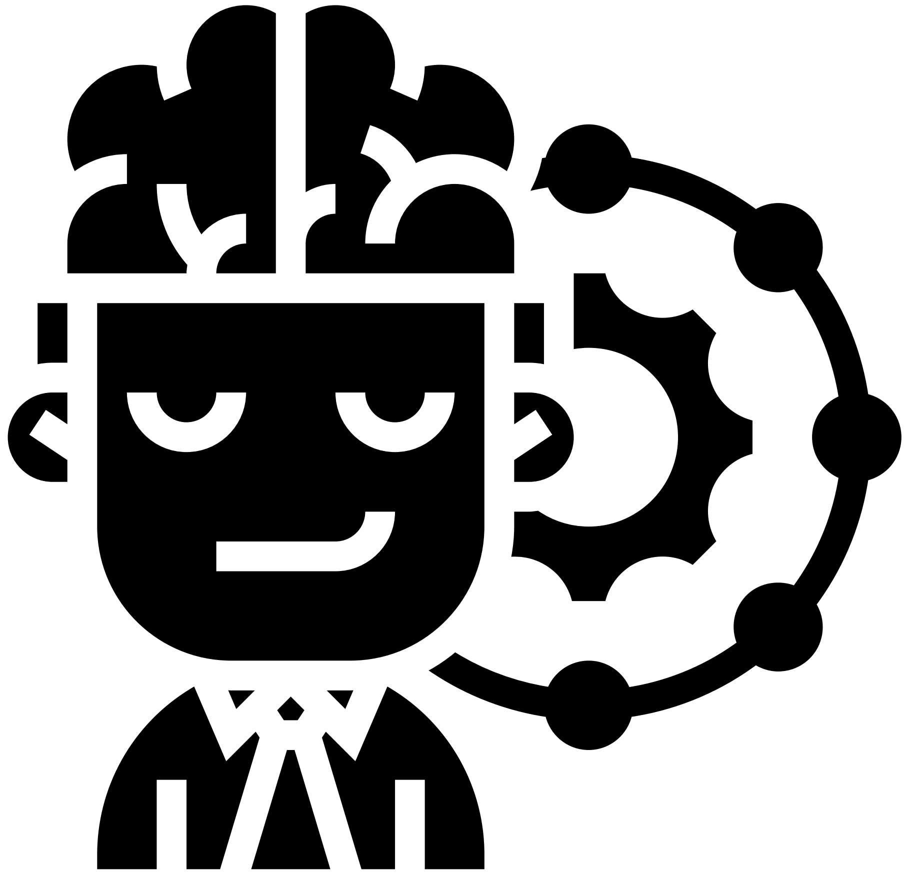
Machine Learning Project



Classification & Regression models

Authors: Fillip Szymański & Zuzanna Miazio

ML process flow

- Initial data inspection & cleaning
- Exploratory Data Analysis
- Feature engineering & preprocessing
- Feature selection
- Models training, validation and selection

Regression

Red Wine Quality

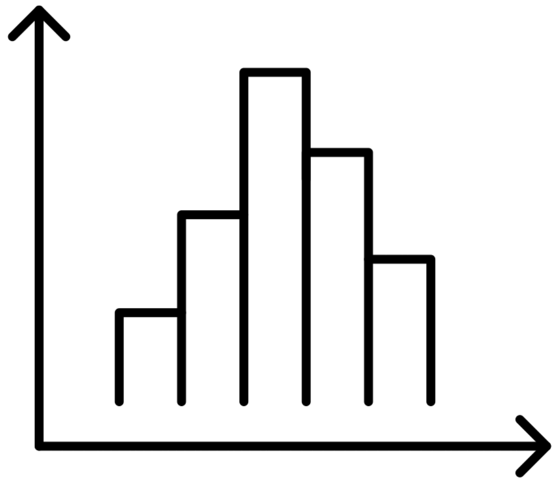
The dataset is related to the Portuguese "Vinho Verde" wine.
It consists of:

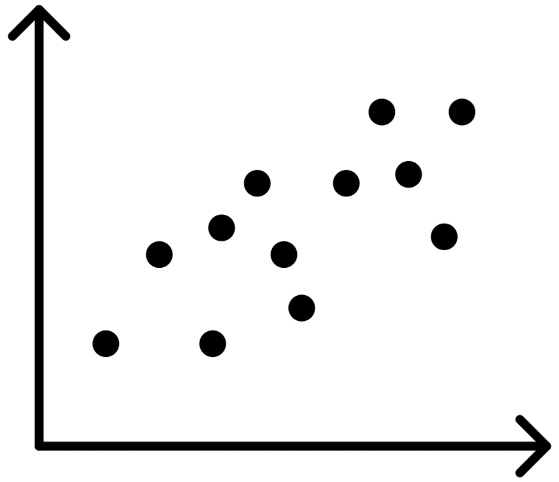
- 20 numeric features (10 unknown)
- 1400 observations
- Continuous target variable (wine quality)

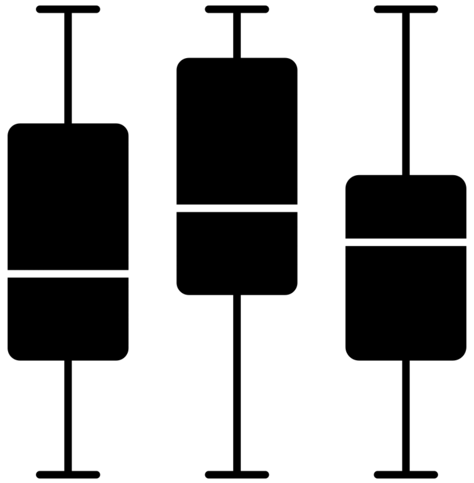
Initial data inspection & cleaning

- Summary statistics
- Data types
- Missing values
- Duplicates

Exploratory Data Analysis

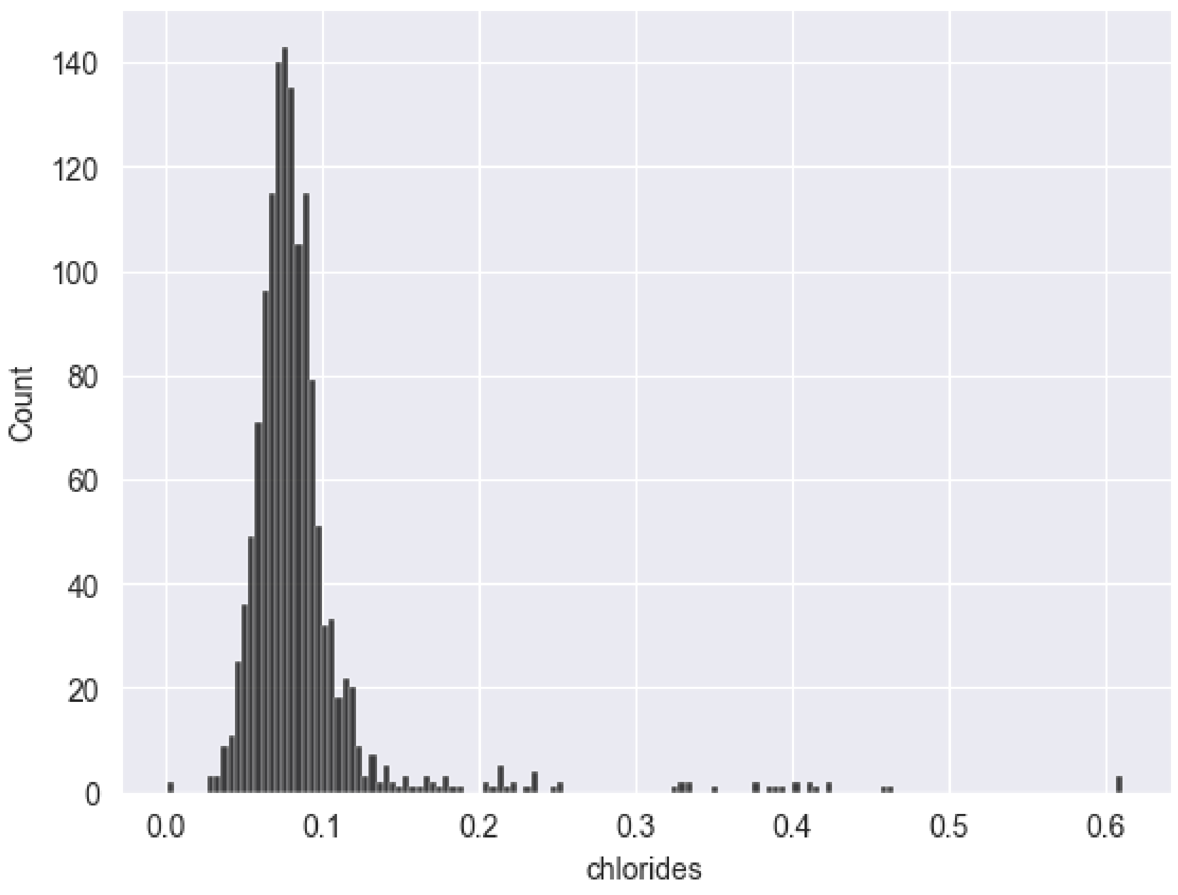


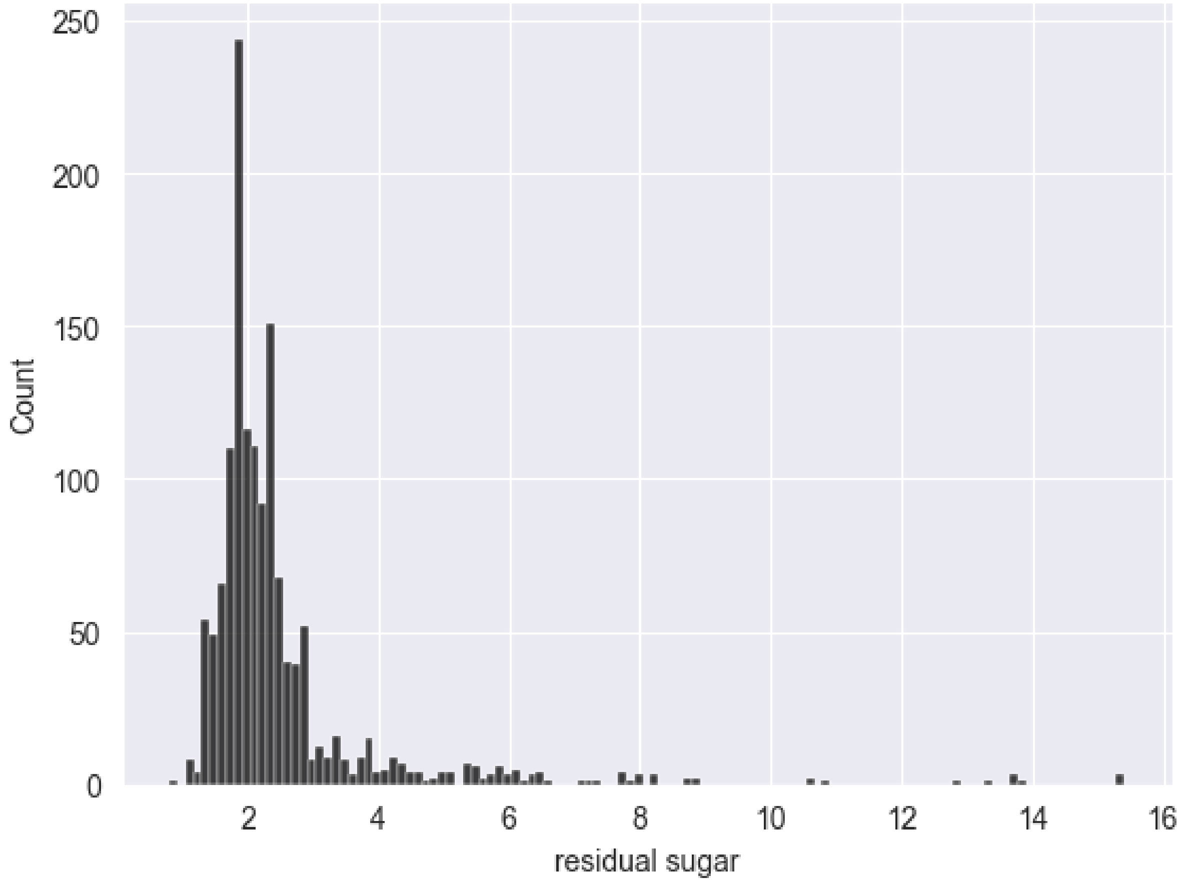


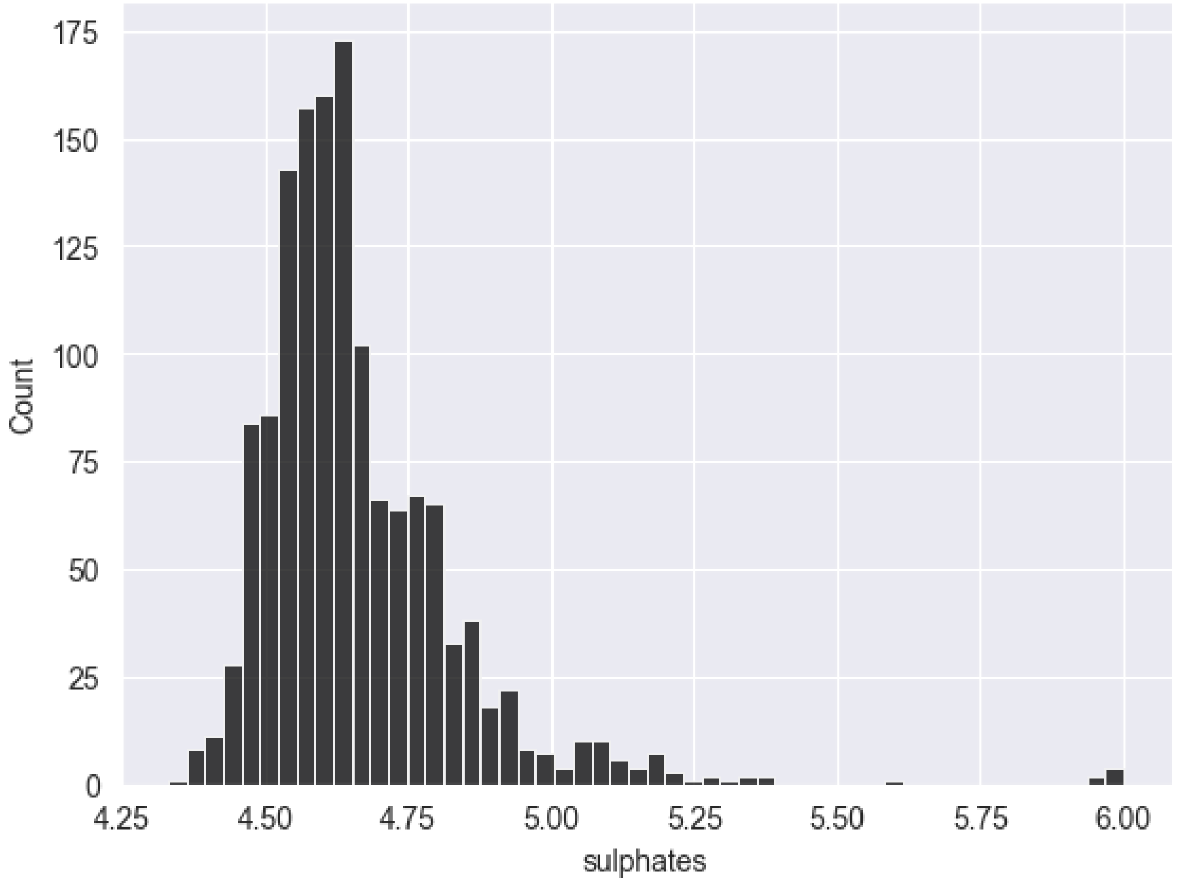


- Histograms
- Boxplots
- Scatterplots
- Correlations between features
- Correlation with target

Feature engineering & preprocessing

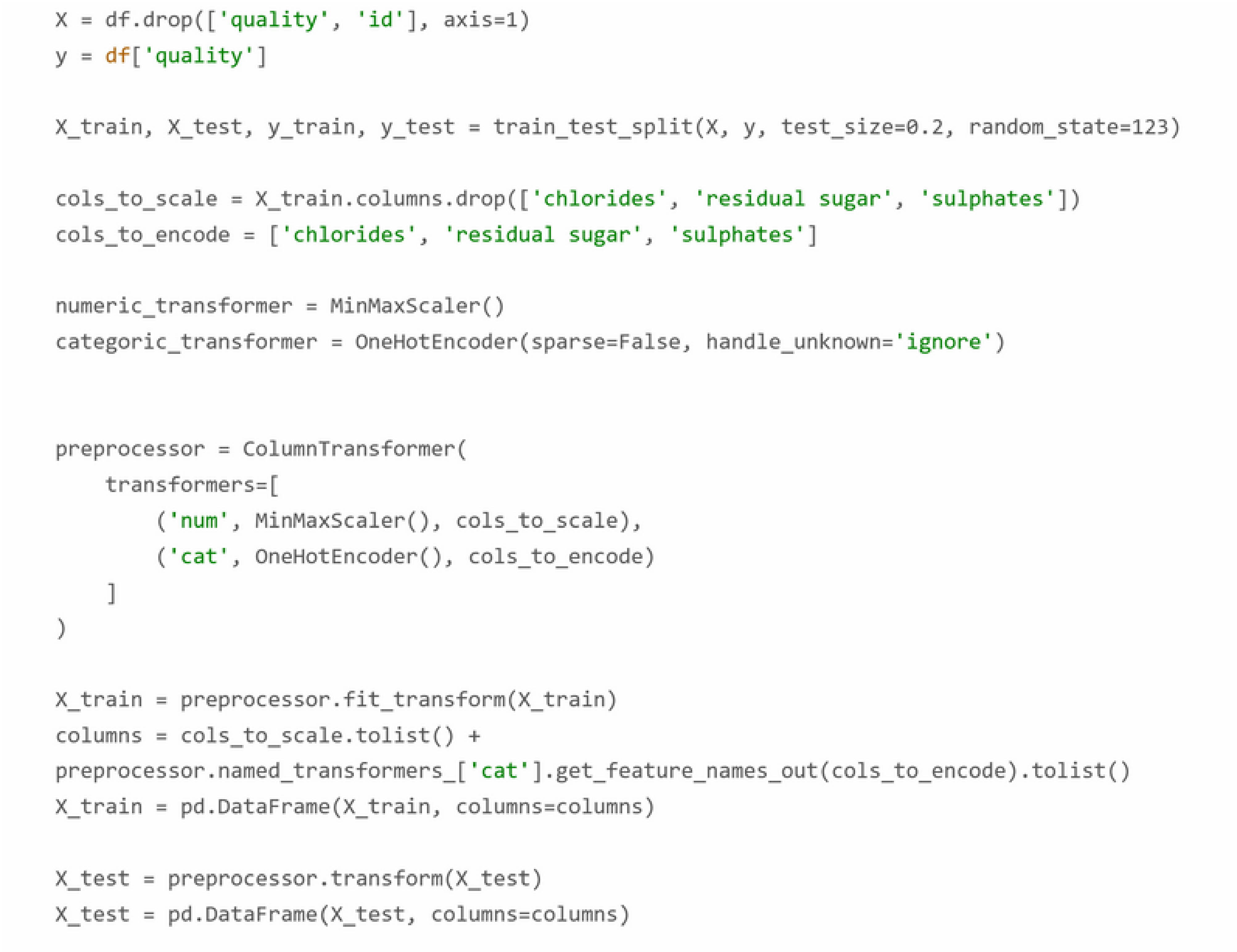






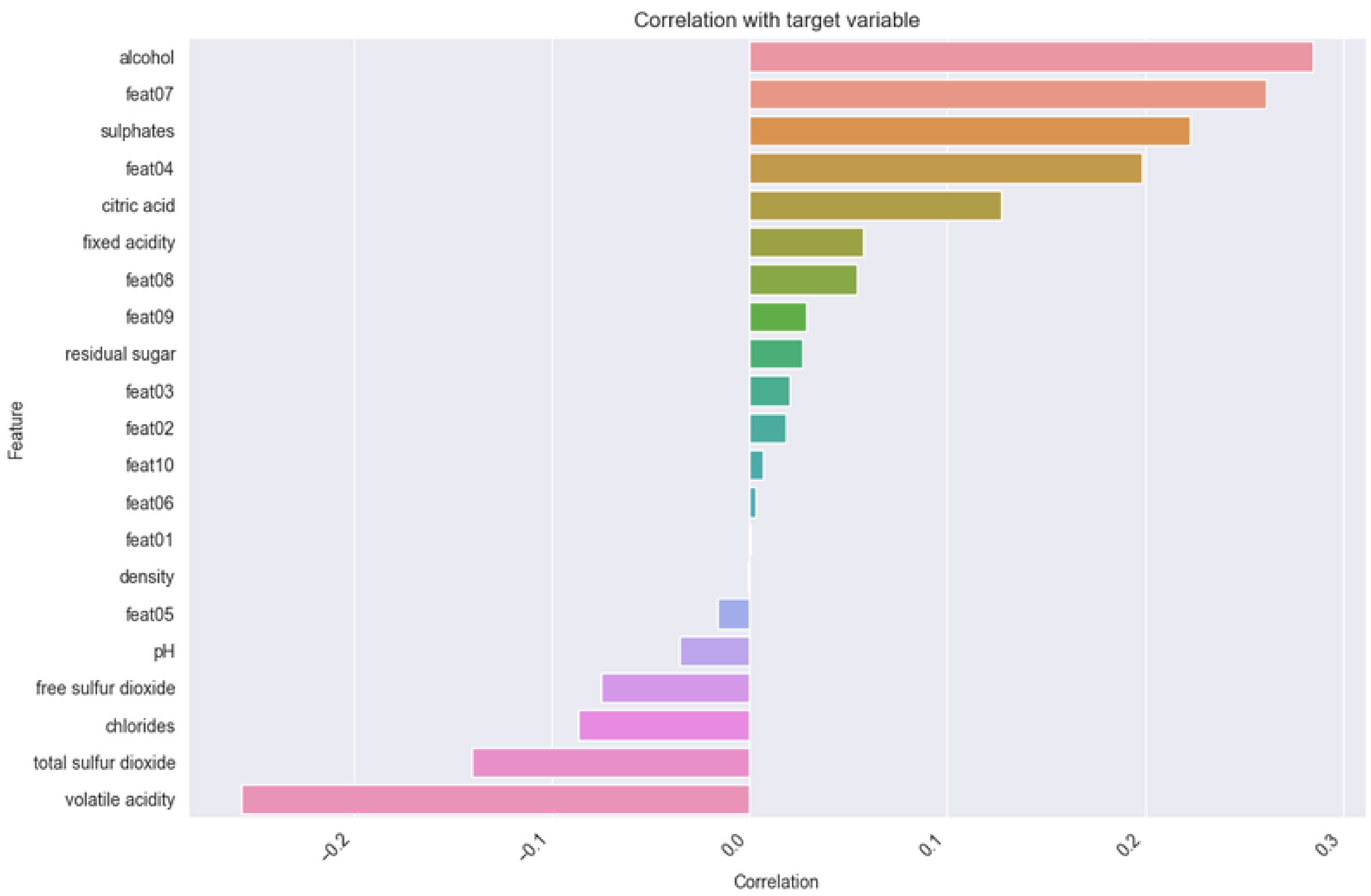
To address outliers in the data, a quantile-based bucketization strategy was employed to categorize values into discrete intervals

Feature engineering & preprocessing



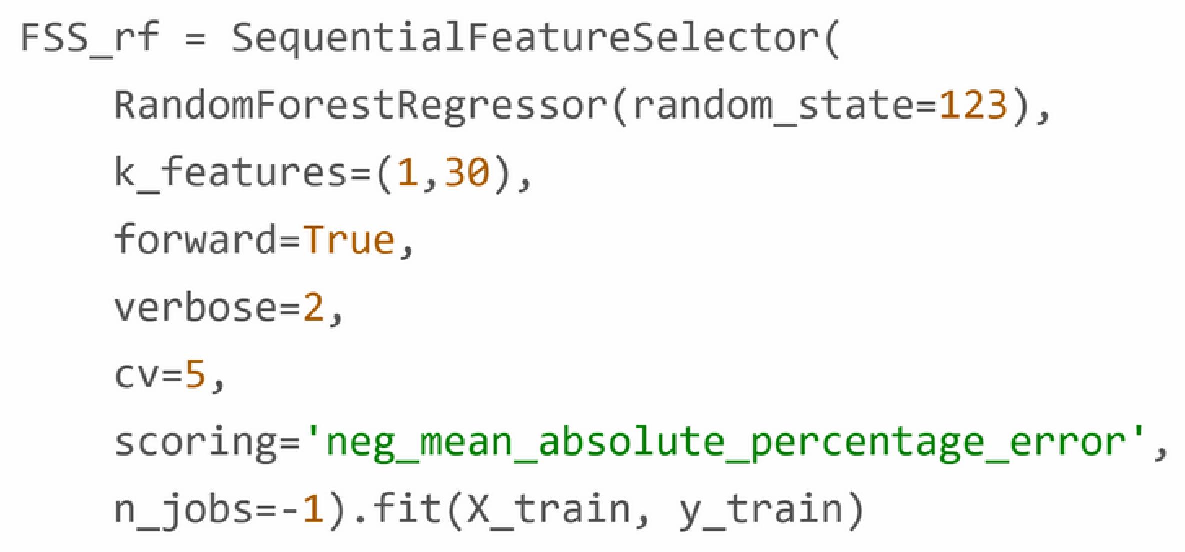
```
X = df.drop(['quality', 'id'], axis=1)
y = df['quality']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=123)

cols_to_scale = X_train.columns.drop(['chlorides', 'residual sugar', 'sulphates'])
cols_to_encode = ['chlorides', 'residual sugar', 'sulphates']

numeric_transformer = MinMaxScaler()
categoric_transformer = OneHotEncoder(sparse=False, handle_unknown='ignore')


preprocessor = ColumnTransformer(
    transformers=[
        ('num', MinMaxScaler(), cols_to_scale),
        ('cat', OneHotEncoder(), cols_to_encode)
    ]
)

X_train = preprocessor.fit_transform(X_train)
columns = cols_to_scale.tolist() +
preprocessor.named_transformers_['cat'].get_feature_names_out(cols_to_encode).tolist()
X_train = pd.DataFrame(X_train, columns=columns)

X_test = preprocessor.transform(X_test)
X_test = pd.DataFrame(X_test, columns=columns)
```

Feature Selection



Forward Feature Selection has been employed to pick only the most relevant features. Consequently, the following variables have been eliminated:

- feat 02, 03, 05, 06, 08, 10
- density



```
FSS_rf = SequentialFeatureSelector(
    RandomForestRegressor(random_state=123),
    k_features=(1,30),
    forward=True,
    verbose=2,
    cv=5,
    scoring='neg_mean_absolute_percentage_error',
    n_jobs=-1).fit(X_train, y_train)
```

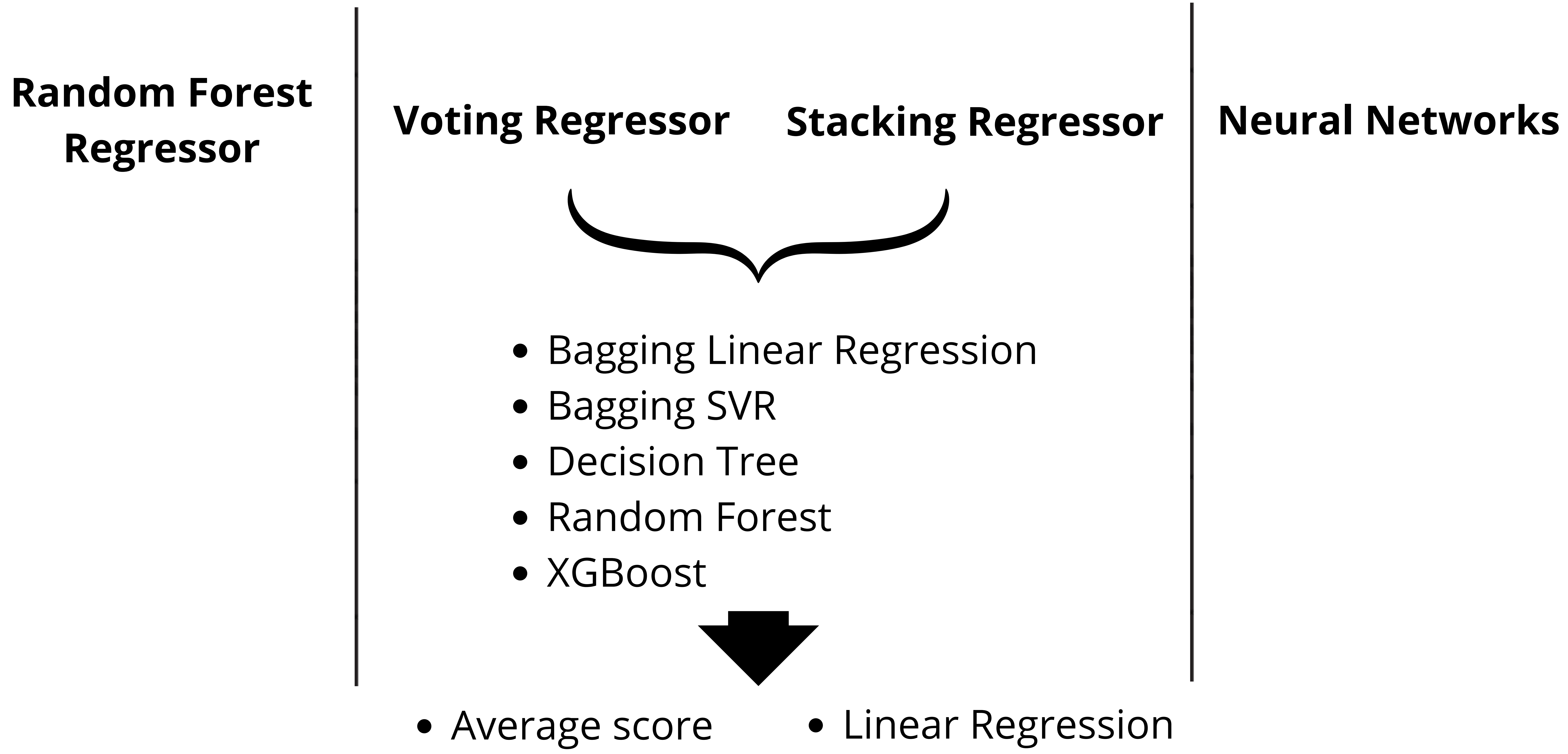
Models training and validation

Four different model configurations have been considered

- Random Forest Regressor
- Voting Regressor
- Stacking Regressor
- Neural Networks

Each of the model's hyperparameters has been fine-tuned using randomized search with cross validation

Models



Random Forest Regressor

Voting Regressor

Stacking Regressor

- Bagging Linear Regression
- Bagging SVR
- Decision Tree
- Random Forest
- XGBoost

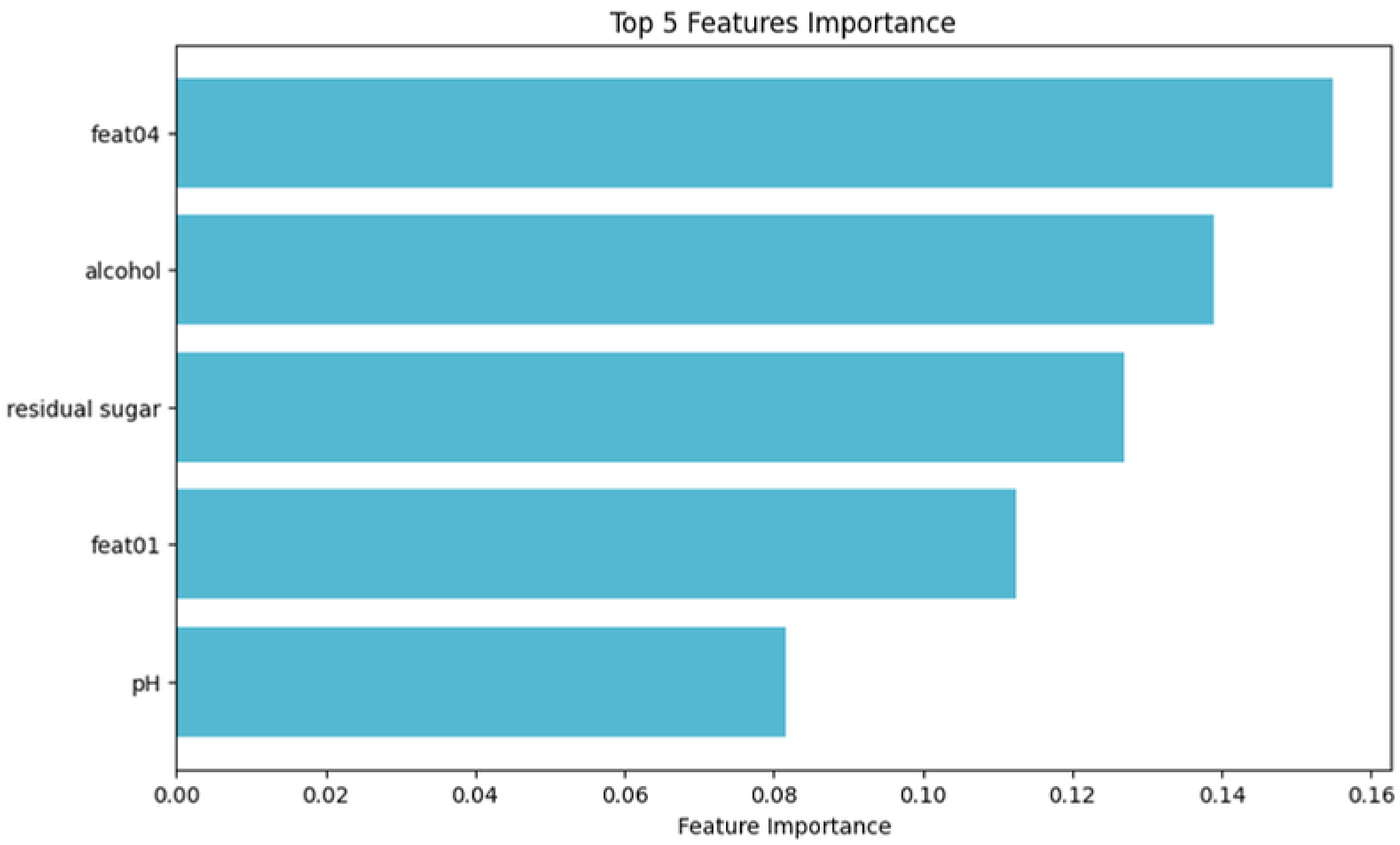
- Average score
- Linear Regression

Neural Networks

Model results

	Random Forest Regressor	Voting Regressor	Stacking Regressor	Neural Networks
Mean Absolute Percentage Error	16.54%	16.85%	16.52%	15.08%
Root Mean Squared Error	1.06	1.08	1.06	1.10
R2	0.24	0.22	0.24	0.08

Features importance



Classification

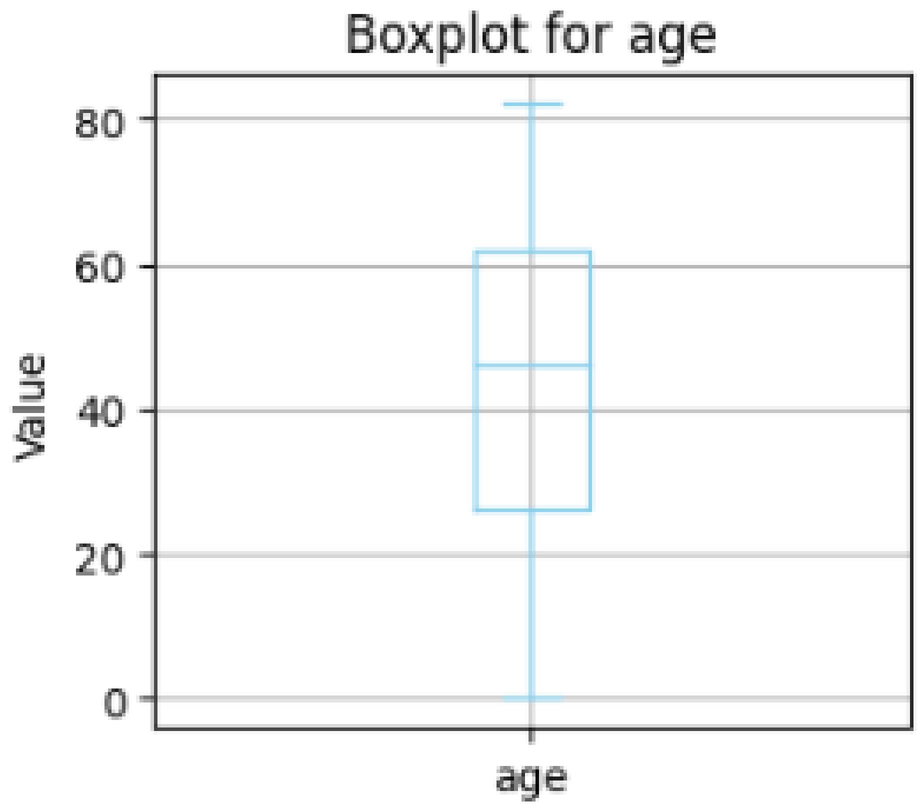
Stroke Prediction

The dataset consists of clinical features for predicting stroke events:

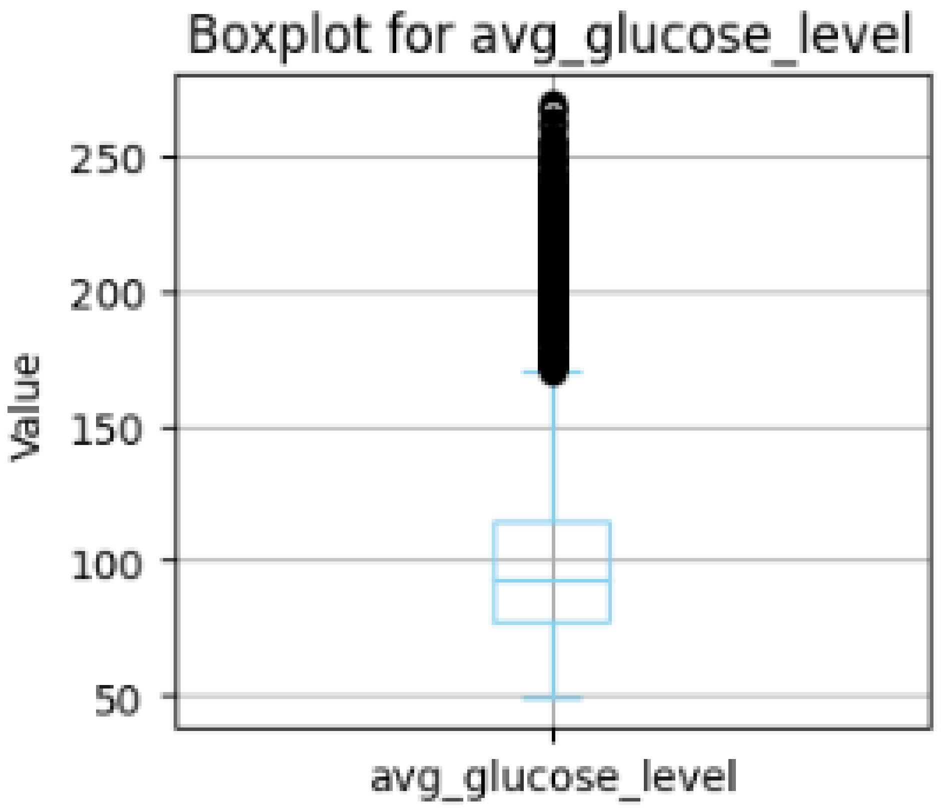
- 7 categorical features describing health and other features of a patient
- 14 numerical features, 10 of which are unknown

Exploratory Data Analysis

- Distribution analysis
- Outlier detection
- Boxplots
- Histograms
- Correlation matrices



Boxplot for age



Boxplot for avg_glucose_level

Data preparation

- Scaling
- Imputation of missing values

Feature engineering and selection

- Feature binning based on quartiles
- One-Hot-Encoding
- Sequential Forward Selection

Models used

Neural Networks

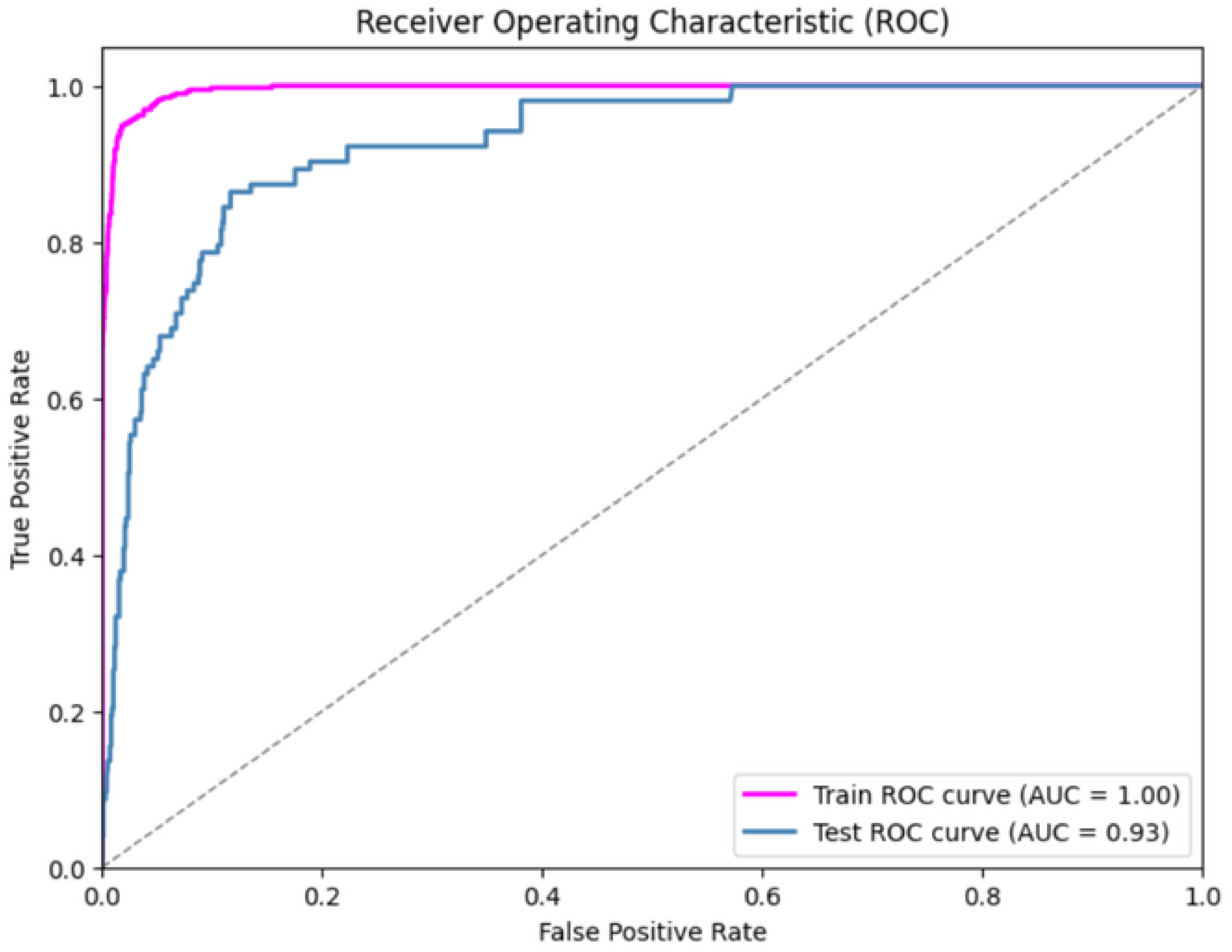
XGBoost

Random Forest

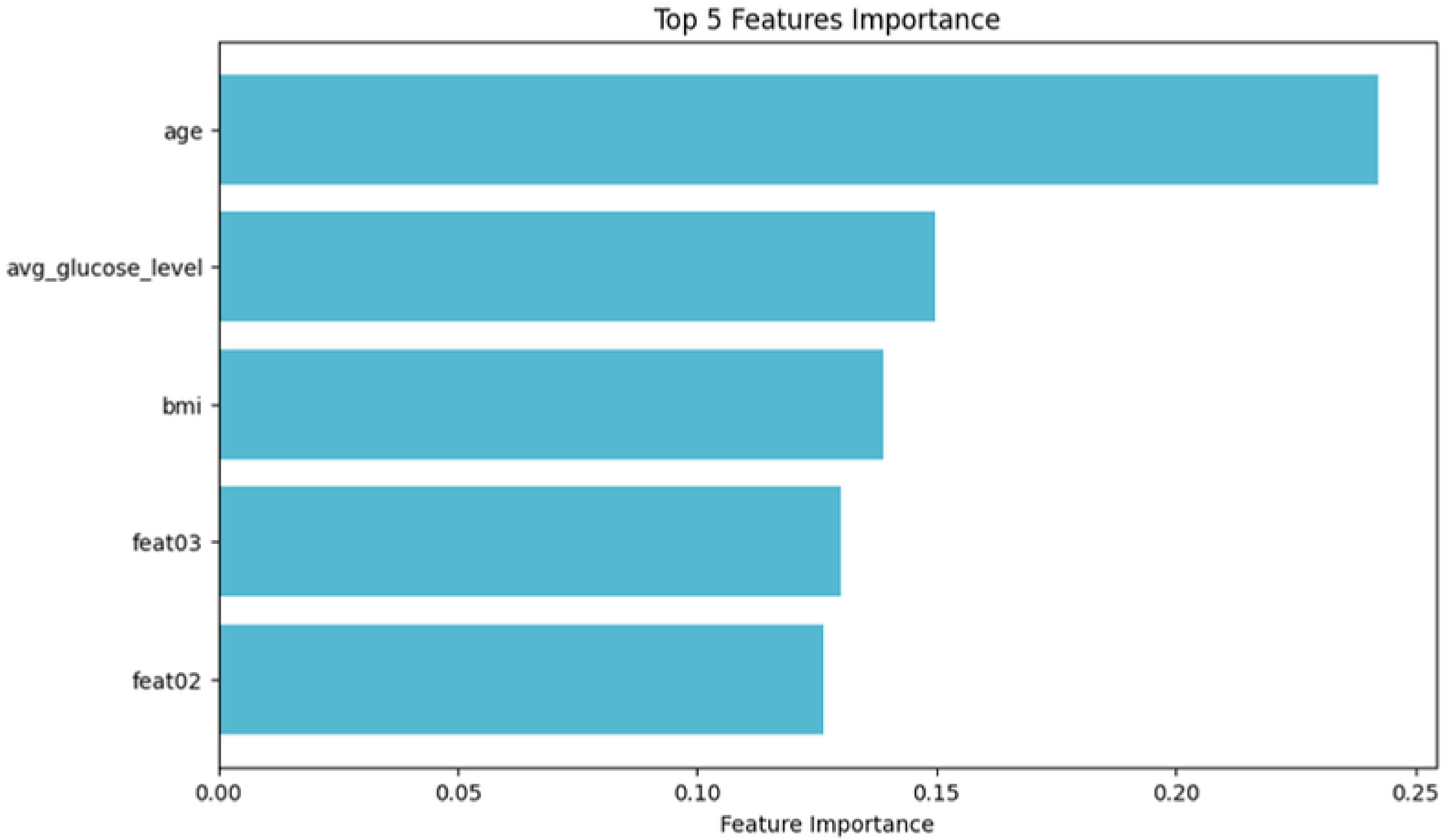
Model results

	Random Forest Classifier	XGBoost Classifier	Neural Networks
Accuracy	0.94	0.92	0.91
AUC	0.95	0.93	0.67
Gini	0.9	0.85	0.34

ROC-AUC curve



Top 5 most important features



Thank you for your attention!